

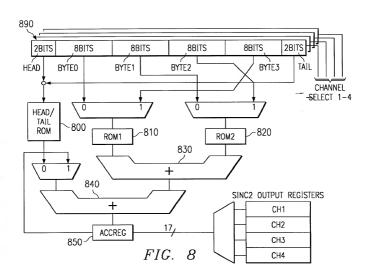
FIFTH ORDER DECIMATE BY 8:

$$H(z) = \left(\frac{1-z-8}{1-z-1}\right)^5$$

36 TAP FIR FILTER. HALF OF THE (SYMMETRIC) COEFFICIENTS

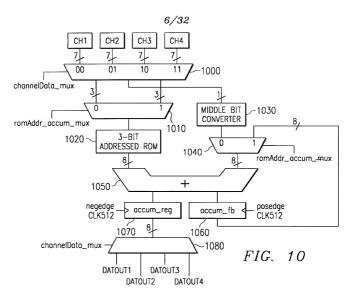
FIG. 7

$h_7 = 330$ $h_8 = 490$	h ₁₆ =2380 h ₁₇ =2460
$h_6 = 210$	h ₁₅ =2226
$h_5 = 126$	h ₁₄ =2010
h ₄ =70	h ₁₃ = 1750
h ₃ =35	h ₁₂ =1470
h2=15	h ₁₁ =1190
h ₁ =5	h ₁₀ =926
$h_0 = 1$	hg =690



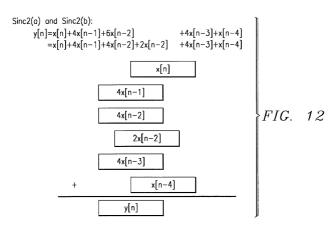
$$H(z) = \left(\frac{1-z^{-2}}{1-z^{-1}}\right)^6$$

IMPULSE RESPONSE: $y[n]=x[n]+6\cdot x[n-1]+15\cdot x[n-2]+20\cdot x[n-3]+15\cdot x[n-4]+6\cdot x[n-5]+x[n-6]$ $FIG. \quad 9$

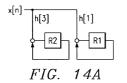


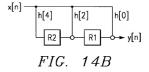
FILTER NAME	SYSTEM FUNCTION	IMPULSE RESPONSE (FILTER COEFFICIENTS)
Sinc2(a) Sinc2(b)	$H(z) = \left(\frac{1 - z^{-2}}{1 - z^{-1}}\right)^4$	h[n]=[1 4 6 4 1]
Sinc2(c)	$H(z) = \left(\frac{1 - z^{-3}}{1 - z^{-1}}\right)^4$	h[n]=[1 4 10 16 19 16 10 4 1]
Sinc2(d)	$H(z) = \left(\frac{1 - z^{-2}}{1 - z^{-1}}\right)^5$	h[n]=[1 5 10 10 5 1]
Sinc2(e)	$H(z) = \left(\frac{1 - z^{-2}}{1 - z^{-1}}\right)^{6}$	h[n]=[1 6 15 20 15 6 1]

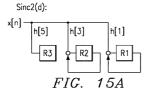
FIG. 11





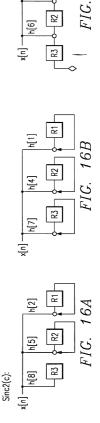


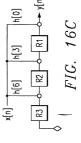


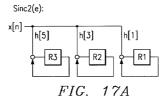


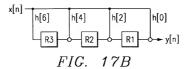
$$FIG. \ \, 13A \left\{ \begin{array}{ll} \sin(2(e)) \\ y[n] = x[n] + 4x[n-1] + 10x[n-2] + 16x[n-3] + 19x[n-4] + 16x[n-5] + 10x[n-6] + 4x[n-7] + x[n-8] \\ = x[n] + 4x[n-1] + \left[\frac{8x[n-2] + 2x[n-2]}{8x[n-2] + 2x[n-2]} + 16x[n-3] + \left[\frac{16x[n-4] + 2x[n-4] + 16x[n-7] + x[n-8]}{8x[n-4] + x[n-4]} \right] \\ + 16x[n-5] + \frac{16x[n-5] + \frac{16x[n-2] + 2x[n-6]}{8x[n-6] + 2x[n-7] + x[n-8]} + \frac{16x[n-3] + \frac{16x[n-3] + 2x[n-4] + x[n-8]}{8x[n-4] + x[n-3]} + \frac{16x[n-3] + \frac{16x[n-3] + 2x[n-3]}{8x[n-4] + x[n-5]} + \frac{16x[n-3] + 2x[n-3]}{8x[n-4] + x[n-5]} + \frac{16x[n-3] + 2x[n-3]}{8x[n-3] + \frac{16x[n-3] + 2x[n-3]}{8x[n-$$

8/32









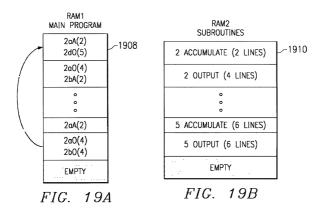


FIG.	18A {	CLK64	$FIG.$ 18A $\{$ case JULULULULULULULULULULULULULULULULULULUL
FIG.	$FIG.$ 18 B { 20A(2)	2oA(2)	
FIG.	FIG. 18C \{ 200(4)	2a0(4)	+ + + + + +
FIG.	FIG. $18D\{$ 2ba(2)	2bA(2)	-
FIG.	FIG. 18E €	2b0(4)	+ + +
FIG.	18F {	2dA(5)	←
FIG.	$FIG.$ 18 $G\{$ 240(5)	240(5)	+
FIG.	FIG. 18H{ 2eA(6)	2eA(6)	_
FIG.	FIG. 18I \ 2e0(6)	2e0(6)	+

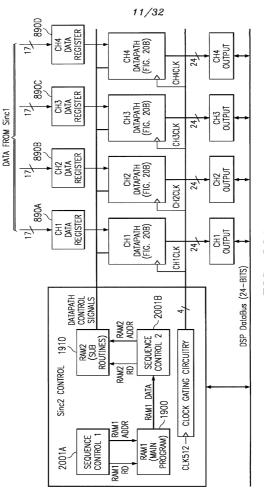


FIG. 20A

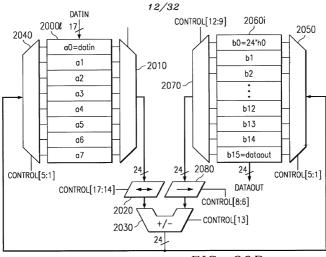


FIG. 20B

PROGRAMMING PROCEDURE:

- 1. SELECT DECIMATION RATE.
- SELECT REQUIRED MINI-SINCS AND ASSOCIATAED ACCUMULATE AND OUTPUT SUBROUTINES.
- 3. SEPARATE COEFFICIENTS INTO FORM SUITABLE FOR SHIFT-ADD OPERATIONS.
- 4. CHECK FOR OVERFLOW AFTER EACH ADDITION IN THE FILTER.
- PERFORM NECESSARY TRUNCATION TO 24 BITS AND SCALING OF SUBSEQUENT COEFFICIENTS IN MINI-SINCS.
- TIME MULTIPLEX ACCUMULATE AND OUTPUT SUBROUTINES SO THAT A MAXIMUM OF 8 ADDITIONS/SUBTRACTIONS ARE PERFORMED FOR EACH INPUT FROM SINC1.
- 7. CREATE CODE FOR RAM2 (ACCUMULATE AND OUTPUT SUBROUTINES) IN THE FORM: [Coeff 1] [Src 1] [Src 2] [Dest] [Coeff2] [Done Subroutine]
- 8. CREATE CODE FOR RAM1 (MAIN CONTROL CODE)
 [Line #] [Wait for new data] [Done program]

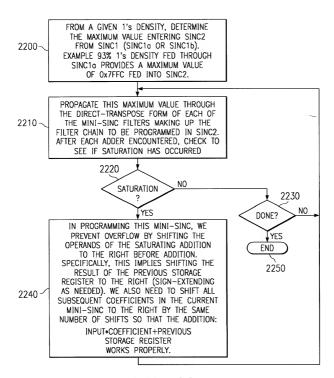
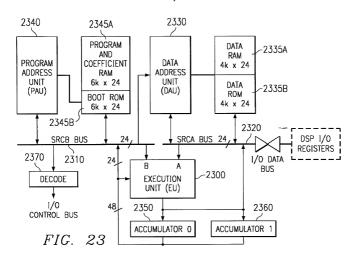
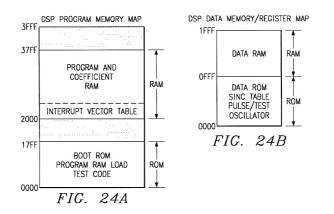
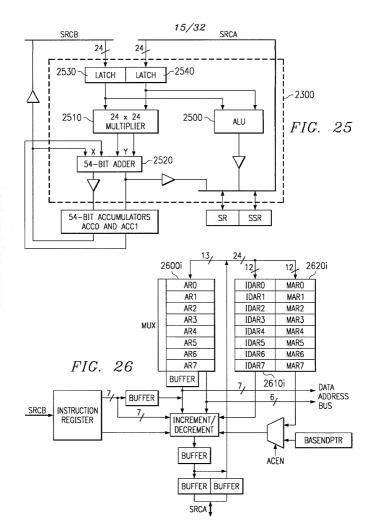
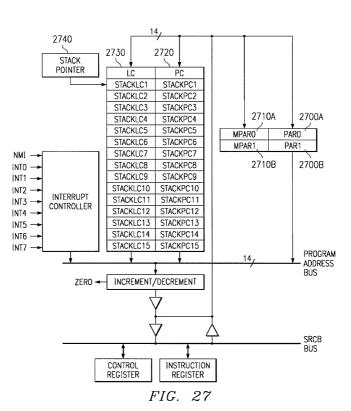


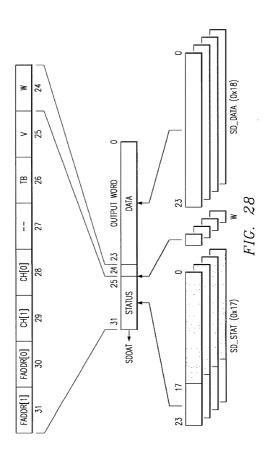
FIG. 22

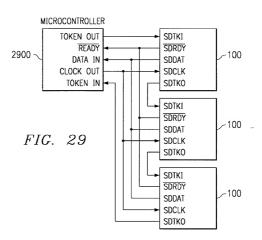












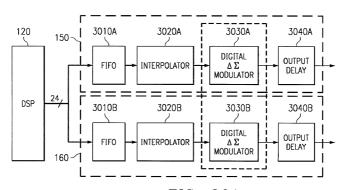
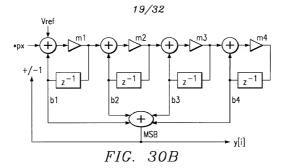


FIG. 30A



$$FIG.$$
 $30C-1$ — wire

 $FIG.$ $30C-2$ — 24 wires

 $FIG.$ $30C-3$ — register

 $FIG.$ $30C-4$ — multiplexer

 $FIG.$ $30C-5$ — tristate buffer

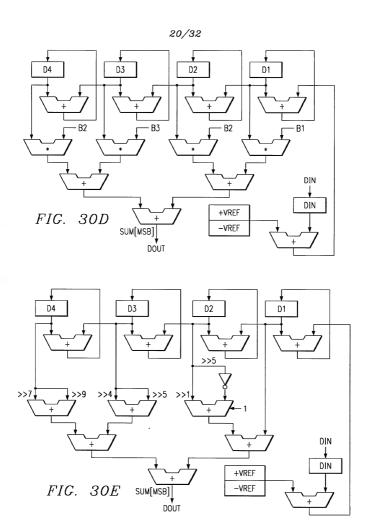
 $FIG.$ $30C-6$ — inverter

 $FIG.$ $30C-7$ — exclusive or gate

 $FIG.$ $30C-8$ — Adder

 $FIG.$ $30C-9$ — Multiplier

 $FIG.$ $30C-10$ — right shifter



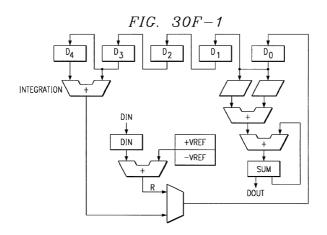
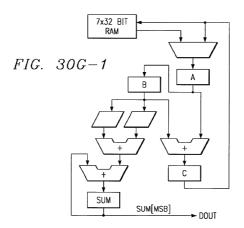
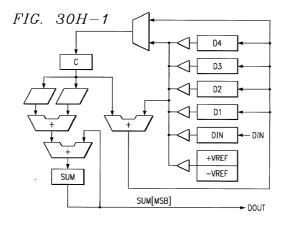


FIG. 30F-2

STATE	ACTIONS DURING STATE								
S0	$D_0(D4_k) = D_4(D4_{k-1}) + D_3(D3_{k-1})$	CLEAR SUM	LOAD DIN _k						
S1	$D_0(D3_k) = D_4(D3_{k-1}) + D_3(D2_{k-1})$	$SUM_k += D_0(D4_k)>>Shift4$							
S2	$D_0(D2_k) = D_4(D2_{k-1}) + D_3(D1_{k-1})$	$SUM_k += D_0(D3_k)>> Shift3$							
S3	$D_0(D1_k) = D_4(D1_{k-1}) + D_3(R_{k-1})$	$SUM_k += D_0(D2_k) >> Shift2$							
S4		$SUM_k += D_0(D1_k)>>Shift1$							
S5	$D_0(R_k) = DIN_k +/- VREF$								





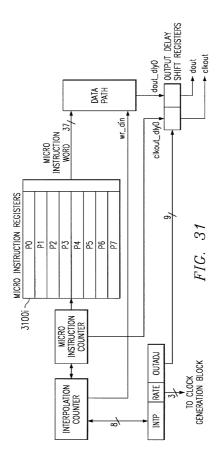
G
O
(1)
U.
1
S.
×
(71
1
F

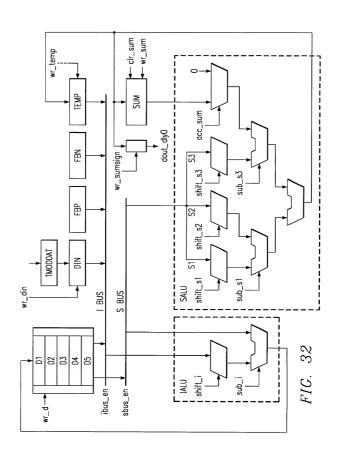
		ACTIONS DURING STATE	ATE	
CLEAR SUM CLEAR C	CLEAR C		CLEAR B	CLEAR A
				LOAD A <mem(d4<sub>k)</mem(d4<sub>
			SHIFT B <a(d4<sub>k)</a(d4<sub>	SHIFT B <a(d4<math>_{\mathbf{k}}) LOAD A<mem(d3<math>_{\mathbf{k}})</mem(d3<math></a(d4<math>
$SUM_k += B(D4_k)>>Shift4 C = B(D4_k) + A(D3_k)$	C = B(D	$^{4}_{k}$) + $^{A}(D3_{k})$	SHIFT B <a(d3<sub>k)</a(d3<sub>	SHIFT B <a(d3<sub>k) LOAD A<mem(d2<sub>k)</mem(d2<sub></a(d3<sub>
				STORE C>Mem(D4 _{k+1})
$SUM_k += B(D3_k)>>Shift3 C = B(D3_k) + A(D2_k)$)B = D	D3 _k) + A(D2 _k)	SHIFT B <a(d2<sub>k)</a(d2<sub>	LOAD A <mem(d1<sub>k)</mem(d1<sub>
				STORE C>Mem(D3 _{k+1})
$SUM_k += B(D2_k) >> Shift2 C = B(D_k)$	C = B(C	$C = B(D2_k) + A(D1_k)$	SHIFT B <a(d1<sub>k)</a(d1<sub>	SHIFT B <a(d1<sub>k) LOAD A<mem(din<sub>k)</mem(din<sub></a(d1<sub>
				STORE C>Mem(D2 _{k+1})
$SUM_k += B(D1_k)>>Shift1 C = B(D1_k) + A(DIN_k)$	C = B($ D1_k\rangle + A(DIN_k)$	SHIFT B <a(din<sub>k)</a(din<sub>	SHIFT B <a(din<sub>k) LOAD A<mem(vref)< td=""></mem(vref)<></a(din<sub>
			SHIFT B <a(vref)< td=""><td>SHIFT B<a(vref) a<c(temp)<="" loadreg="" td="" =""></a(vref)></td></a(vref)<>	SHIFT B <a(vref) a<c(temp)<="" loadreg="" td="" =""></a(vref)>
'+ = J	'+ = J	C = +/- B(VREF) + A(TEMP)		
				STORE C>Mem(D1 _{k+1})

FIG. 30C-2

	LOAD DIN k				
	LOA				
STATE		$D4_{k+1} = C(D4_k) + D3_k$	$03_{k+1} = C(03_k) + 02_k$	$D3_{k+1} = C(D3_k) + D2_k$ $D2_{k+1} = C(D2_k) + D1_k$	$03_{k+1} = C(03_k) + D2_k$ $02_{k+1} = C(02_k) + 01_k$
ACTIONS DURING STATE	LOAD C < D4 _k	LOAD C < D3 _k	LOAD C < D2 _k	LOAD $C < D2_k$ LOAD $C < D1_k$	$\begin{aligned} \text{LOAD C} &< \text{D2}_k \\ \text{LOAD C} &< \text{D1}_k \end{aligned}$ $\text{C(TEMP)} &= \text{C(D1}_k) + \text{DIN}_k$
	SO CLEAR SUM	S1 SUM _k += $C(D4_k)$ >>Shift4 LOAD C < $D3_k$	S2 SUM _k += $C(D3_k)$ >>Shift3 LOAD $C < D2_k$	S2 $SUM_k += C(D3_k)>>Shift3$ LOAD $C < D2_k$ S3 $SUM_k += C(D2_k)>>Shift2$ LOAD $C < D1_k$	S2 $SUM_k += C(03_k)>>Snift3$ LOAD C < D2 _k S3 $SUM_k += C(02_k)>>Snift2$ LOAD C < D1 _k S4 $SUM_k += C(01_k)>>Snift3$ C(TEMP) = C(01 _k) + D1N _k
STATE	SO	S1	S2	\$2 \$3	S2 S3 S4

FIG. 30H-2



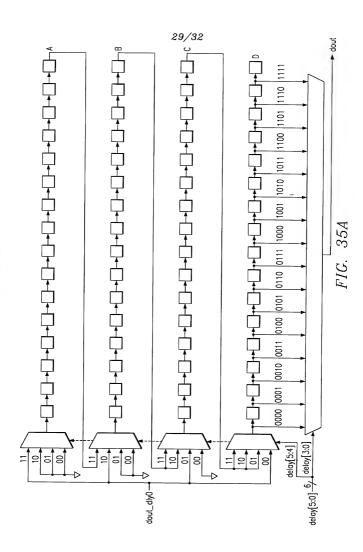


WRITE I	D4	03	02	10	10			
I BUS	+D3	+02	+01	NIQ+	+FB			
S BUS	+D4>>7 +D4>>9 +D4>>9	+D3>>4 +D3>>5 +D3>>8	-02>>4 +02>>1 -02>>7	+D1 +D1 -D1				
TEMP								
SUMSIGN TEMP				WRITE				
SUM	WRITE	ACC./ WRITE	ACC./ WRITE	ACC./ WRITE				
DIN	LOAD DIN _k WRITE							
TEMP								
INTEGRATION	$SUM_k = D4_k > 11$ $D4_{k+1} = D4_k + D3_k$ + $D4_k > 9$ + $D4_k > 7$	$03_{k+1} = 03_k + 02_k$	$02_{k+1} = 02_k + 01_k$	D^{\dagger}_{k+1} ' = D^{\dagger}_k + D^{\dagger}_N	$01_{k+1} = 01_{k+1}$, +/- VREF			
Feedforward	$SUM_k = D4_k >> 11$ + $D4_k >> 9$ + $D4_k >> 7$	$SUM_k = SUM_k + D3_k > 8 + D3_k > 8 + D3_k > 5 + D3_k > 5$	$SUM_k = SUM_k + D2_k > 1$ $= D2_k > 7$ $= D2_k > 7$ $= D2_k > 7$	$SUM_k = SUM_k + D1_k$				
۵	0	-	2	3	4	5	9	7

FIG. 3

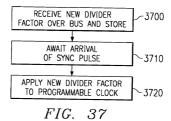
Ð
3
CHE COLUMN
4.0
d G
G
1
e e e e e e e e e e e e e e e e e e e
in in
4

									Ž	28/	32								
	Æ	_			051C 00E4AC	04D3 0082A1	048A 00909F	1469 000003	00000 6200	00000 00000	00000 00000	00000 0000							
						9	04	_		1									
	⋖			0	0	_	-	-	0	0	0	0							
	တ	s s		-	0	0	-	-	0	0	0	0							
				7	-	0	-	0	0	0	0	0							
	S			3	-	0	1	0	0	0	0	0							
	shift_s3		L_	4	0	0	_	0	0	0	0	0							
	S			5	-	-	0	0	0	0	0	0							
				9	0	0	0	0	0	0	0	0		_			_		
				7	1	-	Ţ.	0	0	0	0	0							
	.s2			∞	0	0	0	0	0	0	0	0							
	shift_s2			თ	0	-	0	0	0	0	0	0							
	S		_	0	1	0	0	0	0	0	0	0							
			-	_	0	0	0	0	0	0	0	0	 						
	S	s ←	_	2	0	0	1	0	0	0	0	0							
А			_	~	-	0	0	0	0	0	0	0							
4	S	J	-	4	-	0	0	0	0	0	0	0							
רי.	shift_s1		ļ	5	-	-	-	0	0	0	0	0							
•	S		-	9	0	0	0	0	0	0	0	0							_
5			-	7	0	0	0	0	0	0	0	0							
FIG. 34A	S		1	∞	0	0	0	0	0	0	0	0							
1			-	6	0	0	0	0	0	0	0	0							
	٠-,		2	0	0	0	0	0	0	0	0	0		_	_				
	shift_i		2	-	0	0	0	0	0	0	0	0		_					\vdash
	S		2	2	0	0	0	0	0	0	0	0						L	
	_		2	~	0	0	0	0	0	0	0	0						_	
	tpus_en sbus_en	اوا ا	2	4	0	-	0	-	-	0	0	0						_	
	pns		2	5	0	-	_	0	0	0	0	0				_			
	S		2	9	1	0	0	0	0	0	0	0			_				_
	e,		2	7	-	0	-	1	1	0	0	0							
	bus.		2	∞	_	-	0	0	1	0	0	0				_			
	=		2	6	0	0	0	-	1	0	0	0			_				
	P		~	0	0	1	0	-	-	0	0	0				_			
	wr_d		3	-	0	1	_	0	0	0	0	0							
			3	7	-	0	0	0	0	0	0	0							
	S		3 3	~	0	0	0	0	0	0	0	0			_	<u> </u>			<u> </u>
			_	4	_	1	-	_	0	0	0	0			<u> </u>	_			
	ပ ၁		3	5	0	0	0	0	0	0	0	0			_	<u> </u>			
	S		~	9	0	0	0	-	0	0	0	0				l	ı		ĺ



dout_dly0	DATA OUTPUT BIT, O DELAY
dout	DATA OUTPUT BIT, 0-63 CLOCK DELAY
delay[5:0]	HOW MANY CLOCKS (0-63) TO DELAY OUTPUT DATA dout_dly0
	SELECTS SEGMENT INTO WHICH TO DIRECT dout_dlyO
delay[3:0]	SELECTS WHERE TO TAP SEGMENT D TO GET dout

FIG. 35B



RSU CLOCKS (CREATED FROM CLK16 RISING EDGE)

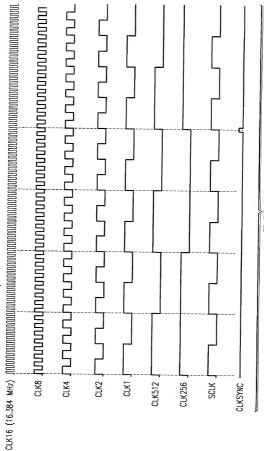


FIG. 36A TO FIG. 36B

